**CUSTOMER CHURN ANALYSIS**

**DEFINITION**

Customer churn or customer attrition is the phenomenon when a company’s customers stop using their product or service during a certain time frame and end their association. A high churn means that a higher number of customers no longer want to purchase goods and services from the business. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. Customer churn can prove to be a roadblock for an exponentially growing organization and a retention strategy should be decided in order to avoid an increase in customer churn rates.

**HOW TO AVOID HIGH CHURN RATES**

New business use marketing and sales budget to gain additional customers. Big businesses have existing customers and they will often have a higher volume of product/service consumption and can generate additional customers through referrals.

A customer can be retained by providing good product and customer service to them. But still many companies have high customer churn rates. So, the most effective way for a company to prevent attrition of customers is by knowing that what customer truly wants. The vast volumes of data collected about customers can be used to build customer churn prediction models so, company can know how many customers are most likely to defect and company can prioritise focused marketing efforts on that subset of their customer base.

**OUR DATASET**

We are provided customer data of telecommunications company from IBM Sample Data Sets.The telecommunications sector has become one of the main industries in developed countries. The technical progress and the increasing number of operators raised the level of competition, Customers’ churn is a considerable concern in service sectors with high competitive services. On the other hand, predicting the customers who are likely to leave the company will represent potentially large additional revenue source if it is done in the early phase

Our Datastes consists customer’s information like their ID, gender, internet service charges, tenure, etc in 21 columns and 7043 rows.

These columns are : -

customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn.

**PROBLEM STATEMENT**

We have to examine the dataset and see if the attrition is present or not. We have to create different machine learning customer churn prediction models and choose the best performing model.

**DATA ANALYSIS**

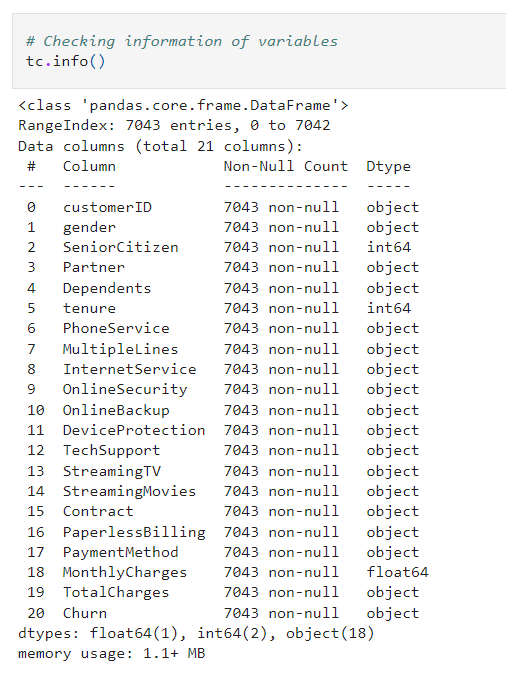
* First we’ll be importing dataset “telecom\_customer\_churn.csv” with the help of pandas library



* Putting Dataset in DataFrame

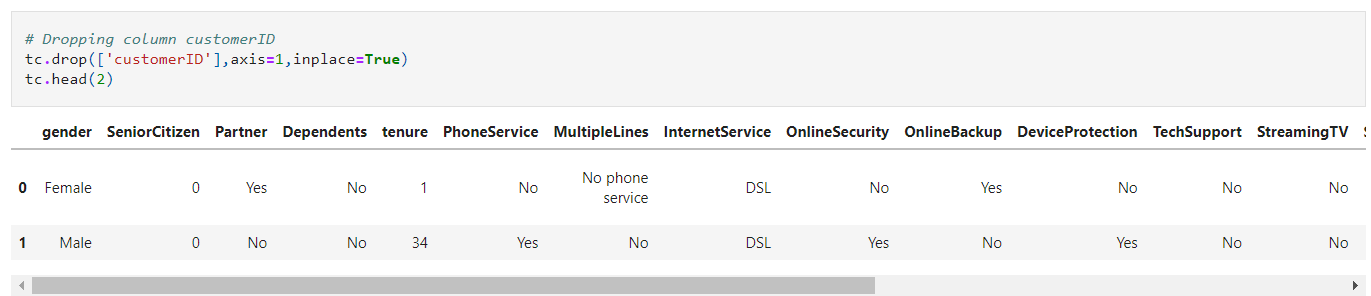


* Checking information of columns



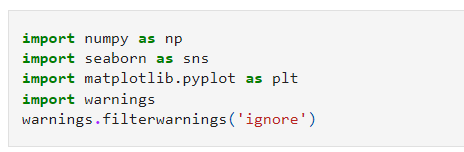
\*\* We haven’t found any null values in the dataset

* Dropping column customerID as it is of no use for our model.

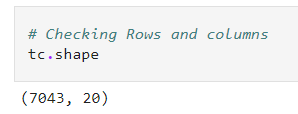


**EXPLORATORY DATA ANALYSIS**

* Importing important libraries



* Shape of Dataset

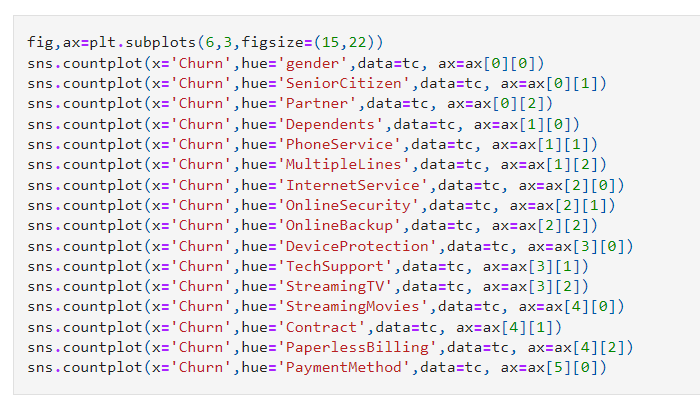


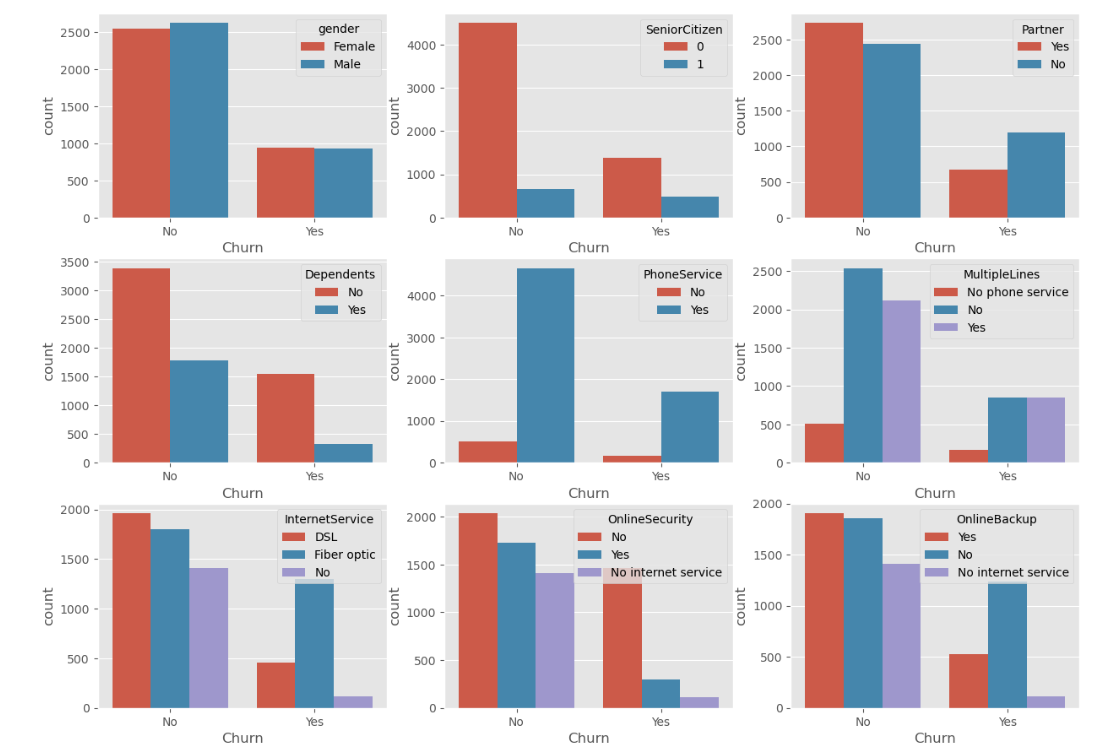
**PLOTS USED : -**

**COUNTPLOT**

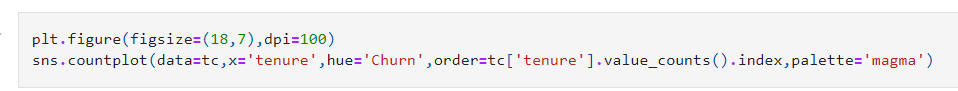
Countplot is a method which is used to show the counts of observations in each categorical bin using bars.

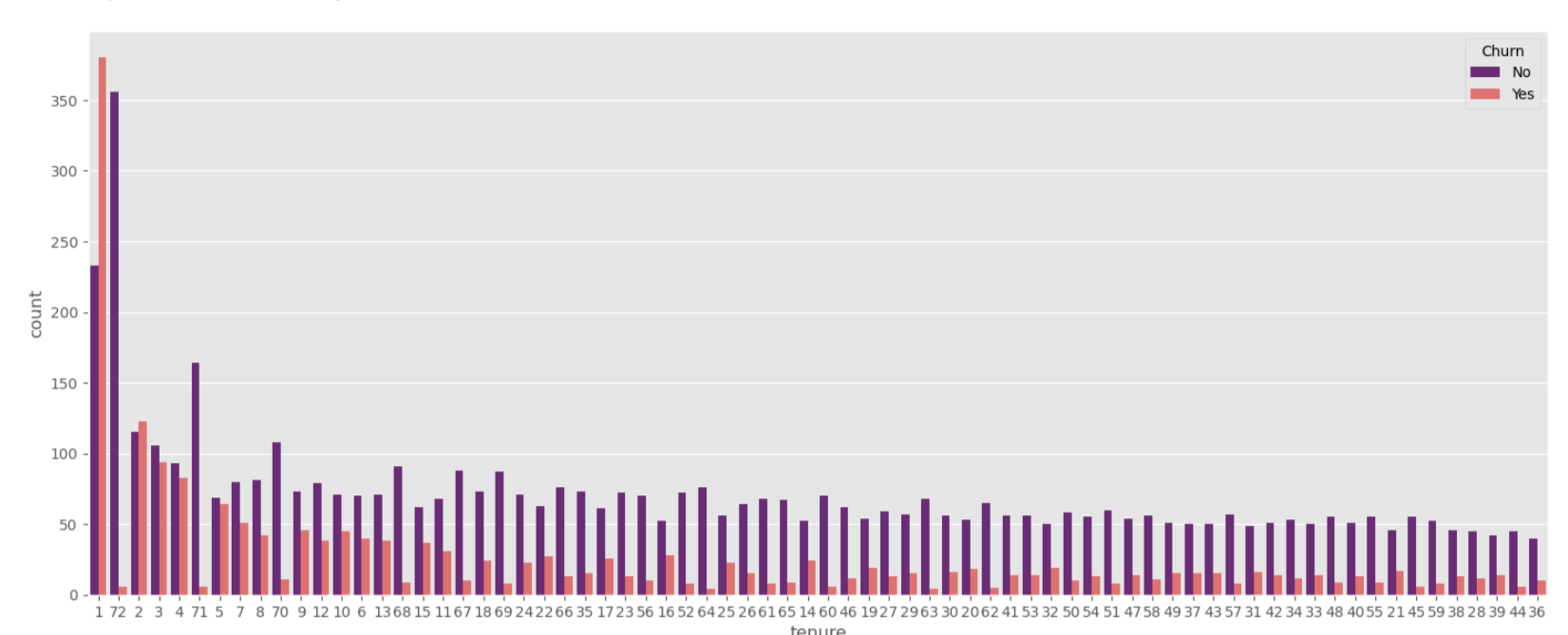
1. We’ve used countplot from seaborn to count the number of people who are going to defect with respect to the different variables. We’ve given variable ‘Churn’ on x-axis and other variables in hue.



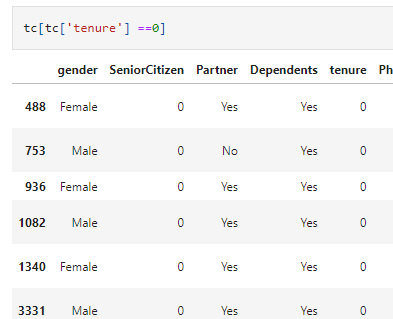


1. We’ll count how many people of different age have yes or no churn. Here we will use countplot, where x is ‘tenure’ with hue ‘Churn’ and we will use ‘magma’ in palette.





* Checking how many rows of variable ‘tenure’ is containing 0

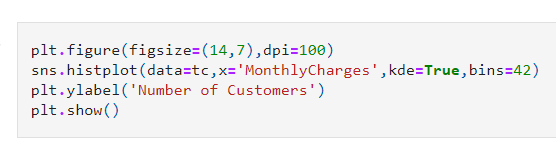


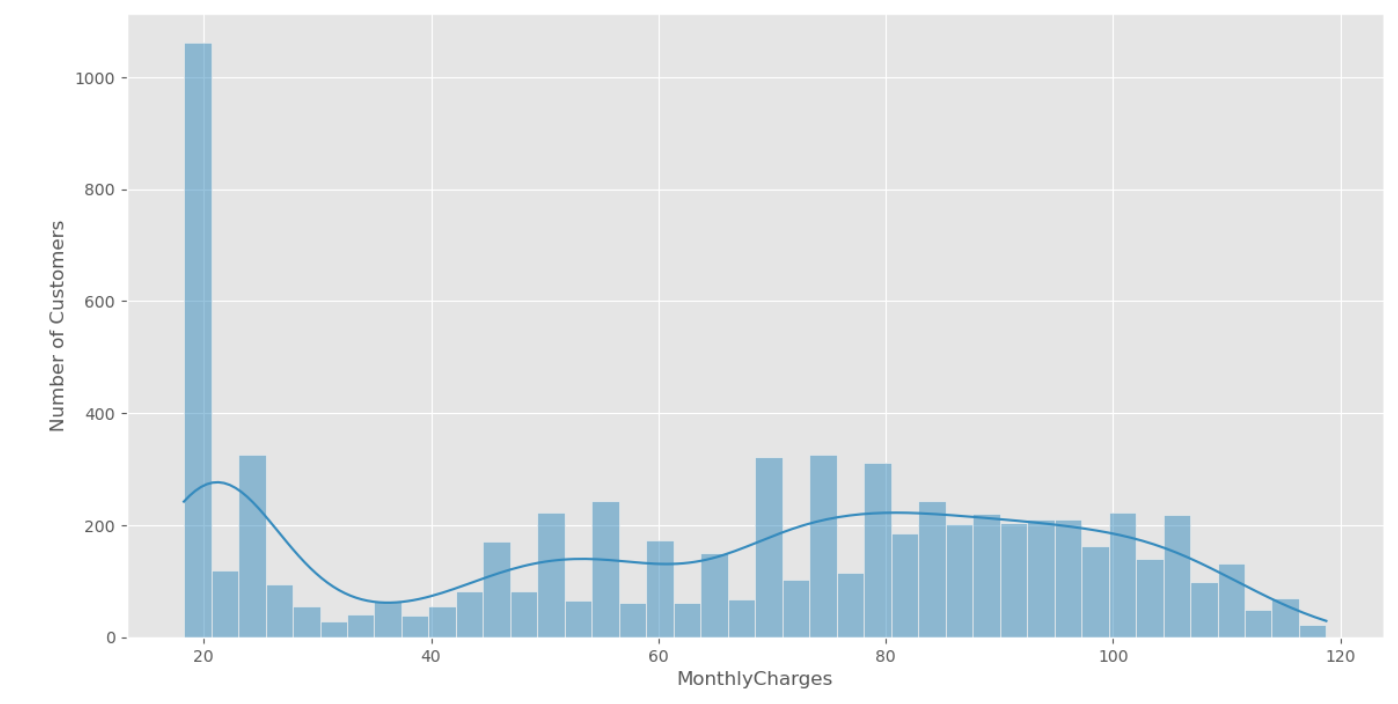
**In the above table tenure is 0 and there is no total charges so we'll drop these rows**

**HISTPLOT**

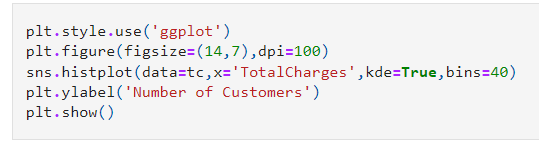
A histogram is a classic visualization tool that represents the distribution of one or more variables by counting the number of observations that fall within discrete bins.

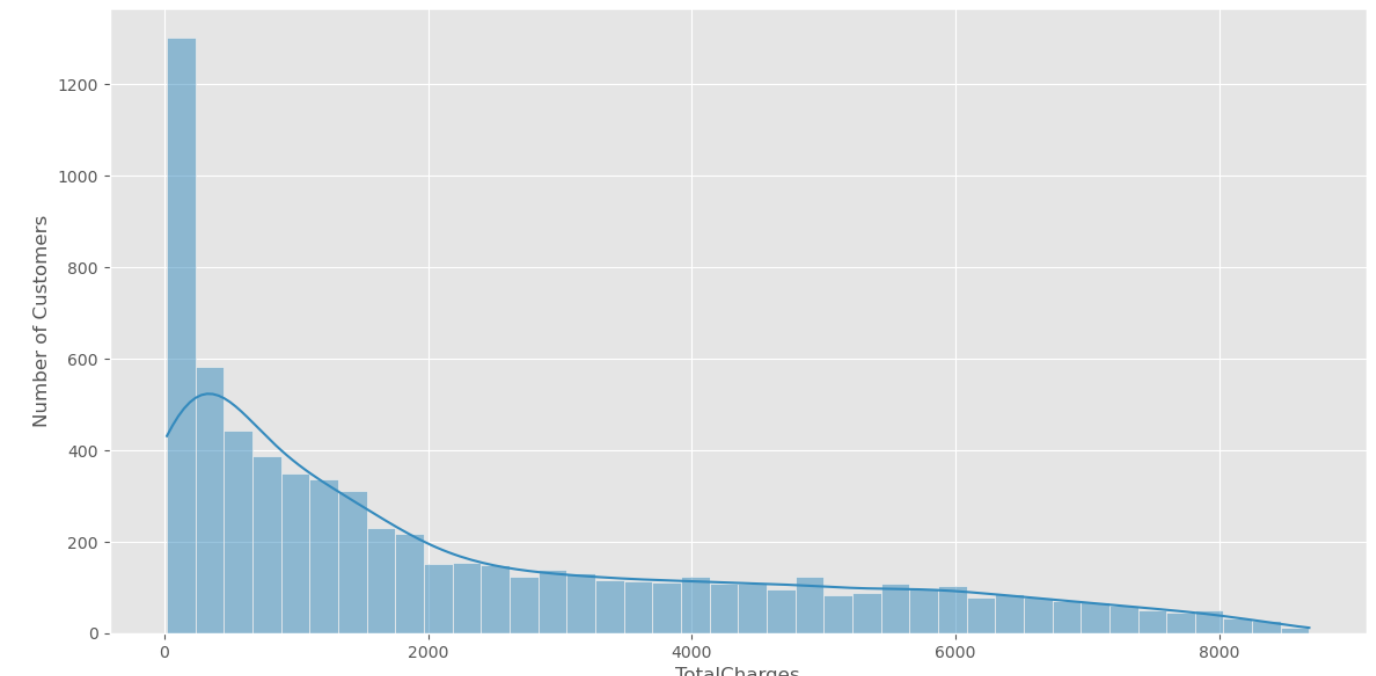
1. We’re using histplot from seaborn to represent distribution of number of customers by variable ‘MonthlyCharges’. Here we’ve given variable ‘MonthlyCharges’ to x-axis and we have given kde = true which means the line showing in the plot.





1. Converting variable ‘TotalCharges’ into float by changing ‘astype(float)’, and using histplot to show the distribution of number of customers by ‘TotalCharges’. Here x-axis have TotalCharges with kde=True



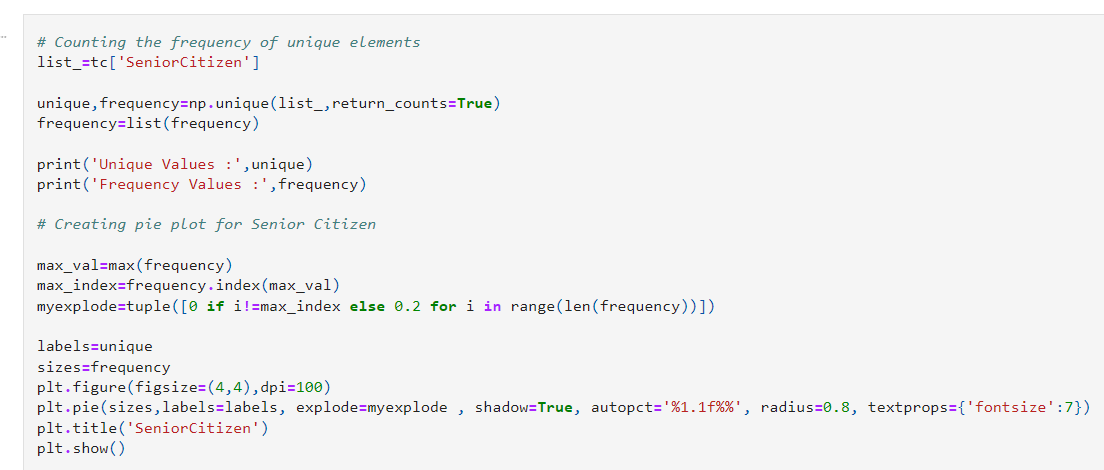


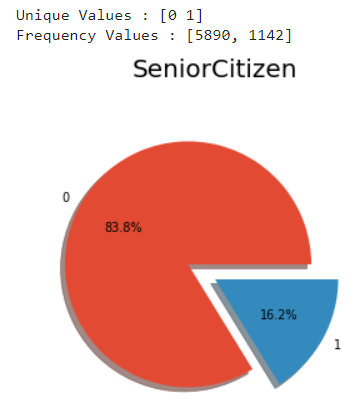
**PIEPLOT**

Step – 1 Creating list for variable to return the counts of unique values and their frequency

Step – 2 Selecting maximum value from frequency and creating variable ‘myexplode’,we will use ‘for’ loop for getting maximum value and explode it by 0.2 from the pie and other pieces will remain intact to the pie.

Step – 3 Creating pie plot, firstly we’ll create variables for parameters like labels and sizes. We’ll make the shadow true and use autopct, it enables you to display the percent value using Python string formatting, autopct='%1.1f%%' means that for each pie wedge, the format string is '1.1f%'.

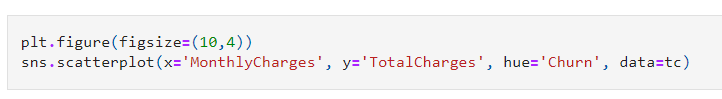


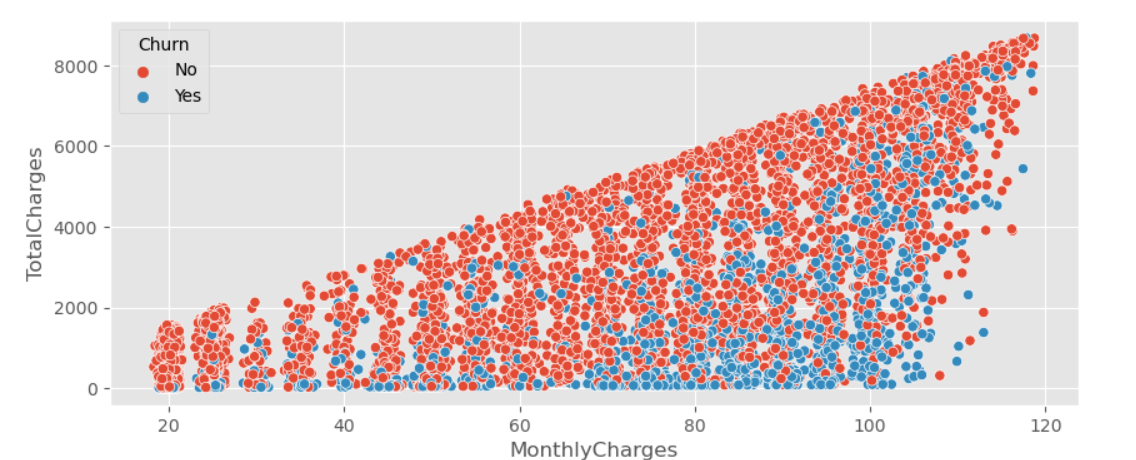


**SCATTERPLOT**

Scatter plots are the **graphs that present the relationship between two variables in a data-set**.

**We’ll see the relationship between ‘MonthlyCharges’ and ‘TotalCharges’ by using scatter plot from seaborn, x-axis – ‘MonthlyCharges’, y-axis – ‘TotalCharges’, hue – ‘Churn’**

****

****

**EXPLORATORY DATA ANALYSIS CONCLUSION**

It is important to know about the data which are we going to use for creation of machine learning models. We have imported the csv file of data and stored it in a DataFrame. We have done some important exploratory data analysis for our data. We know the shape of the dataset is 7043 rows and 20 columns, there is no null values present in the dataset and most of the columns are categorical and very few are integers and float. Data is very big entity and we must see the data visually to understand it better. We have plotted the data in different plots which includes count plot, hist plot, box plot, scatter plot and pie plot to see the visual distribution and relationship among the variables. We’ve noticed that only 16.2 percent people are senior citizen, less than 30% of churn is present, majority of people pay less than 40 and less than 2000 as monthly charges and total charges. There is a direct relationship between Total Charges and Monthly Charges. These methods and plots helped us very much to see through the data and understand it better. Now it will be easy for us to do the feature engineering and creating machine learning models.

**PRE - PROCESSING**

**CONVERTING CATEGORICAL DATA INTO INTEGER**

**Some of the variables are in categorical form and python has some methods which don’t work on categorical data so we have converted them in integer.**

**LABELENCODER**

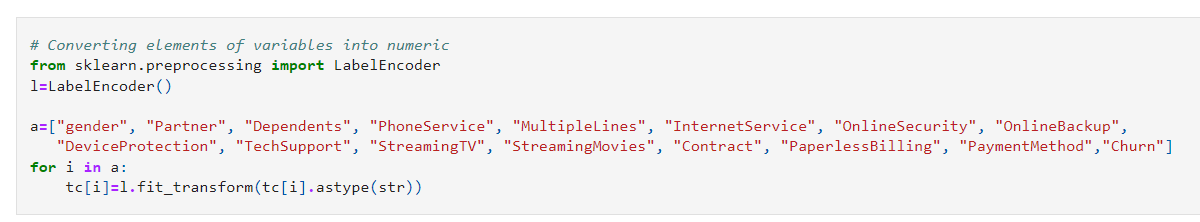
LabelEncoder will convert the categorical elements into numbers like 0, 1, 2, etc.

Step – 1 Import LabelEncoder from library sklearn.preprocessing

Step – 2 Create variable for LabelEncoder

Step – 3 Create list for categorical variables

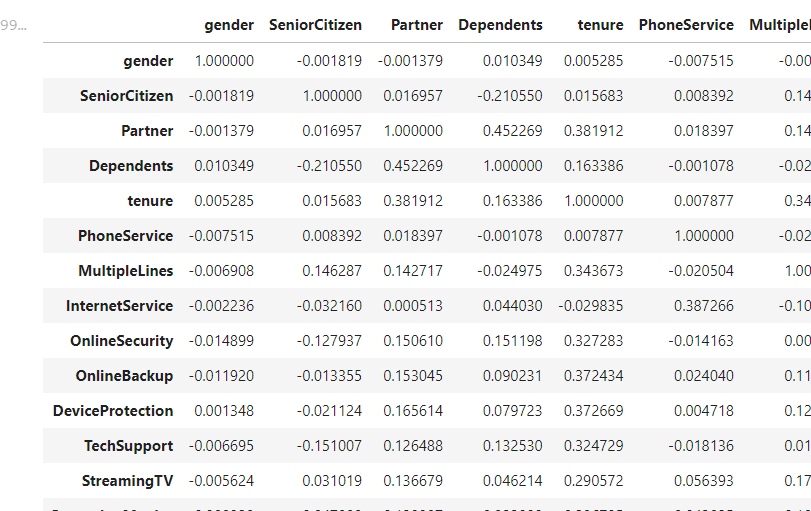
Step – 4 Create a for loop and encode all the categorical variables



\*\* Now all the variables are integers.

**CHECKING CORRELATION AMONG THE VARIABLES**

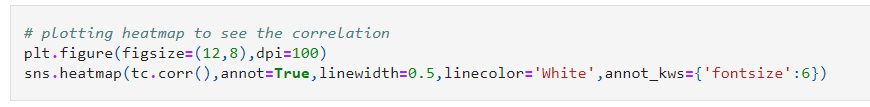


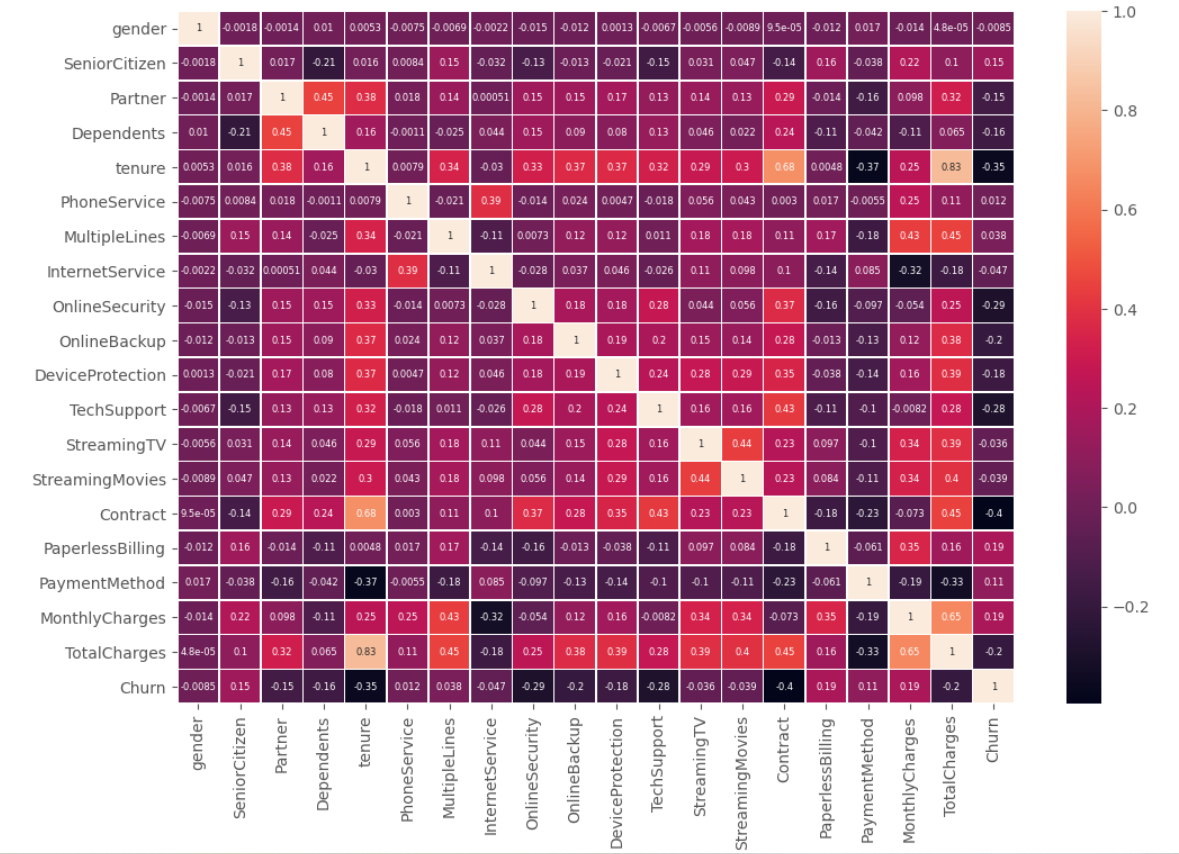


**VISUALIZATION OF CORRELATION**

We’ll use heatmap,  A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade.

We’ll make the annotations true, width between the lines is 0.5 and color of lines are white.





**Key Observations : -**

**Most variables are correlated with each other but not with Churn**

**tenure is highly correlated with contract and Total Charges**

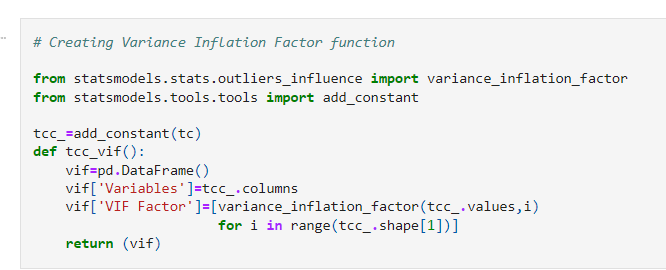
**VARIANCE INFLATION FACTOR (VIF)**

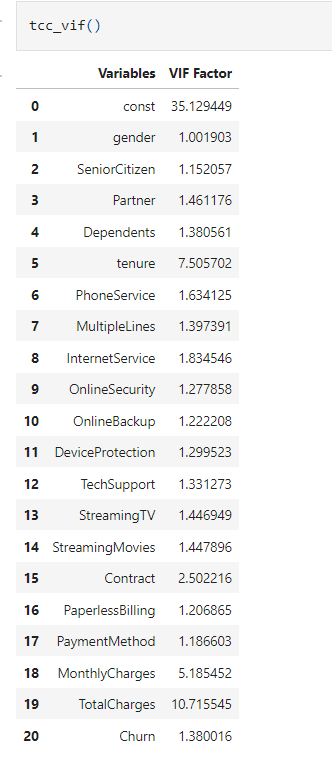
Step – 1 Importing variance\_inflation\_factor from statsmodels.stats.outliers\_influence, Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables.

Step – 2 Import add\_constant from statsmodels.tools.tools, it adds a constant term to the linear equation it is fitting

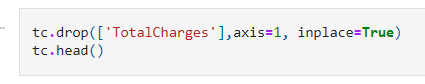
Step – 3 Creating variable to store add\_constant

Step – 4 Creating function named tcc\_vif() for calculating variance inflation factor of all the integer variables and making ‘for’ loop for giving the shape.



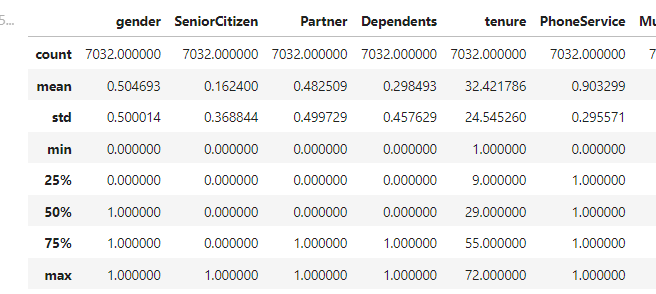


**\*\* Tenure and Total Charges have highest variance inflation factor, we'll drop Total charges as it has the highest inflation factor**



**Checking Statistics of variables like mean, median, standard deviation, etc**



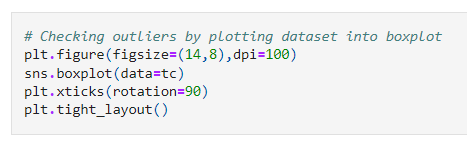


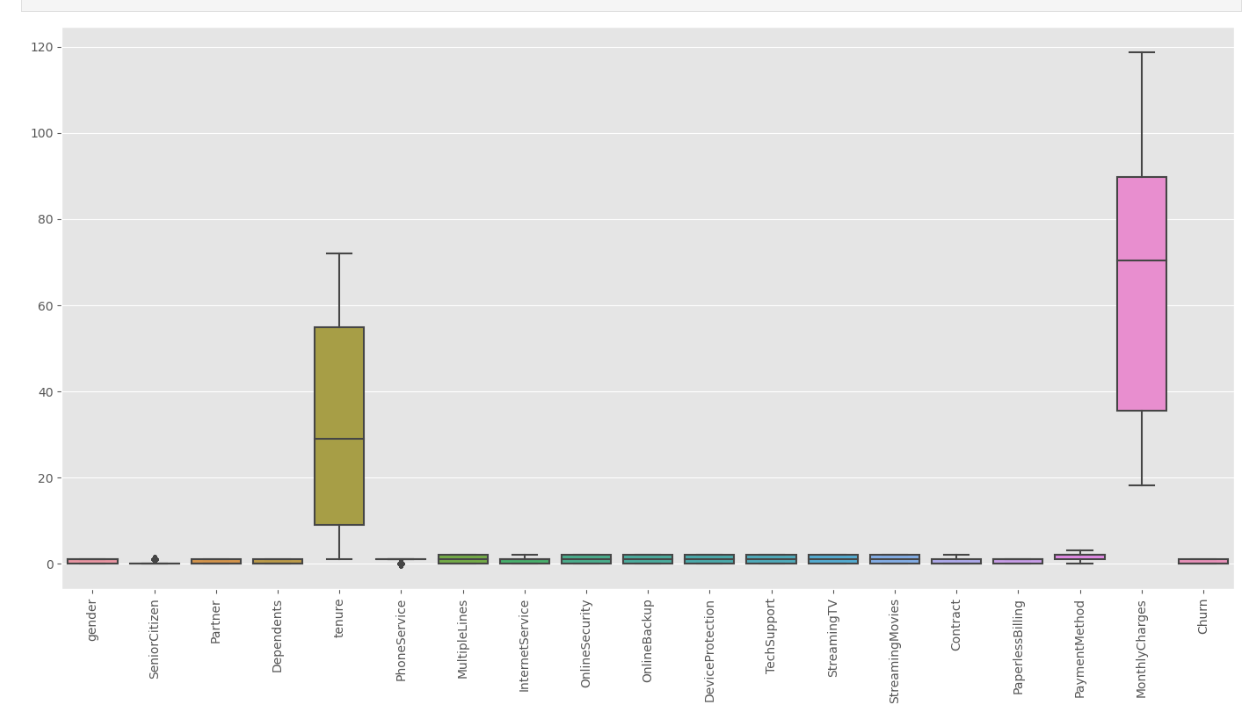
**BOXPLOT**

A **Box Plot** is also known as **Whisker plot**is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

**CHECKING OUTLIERS**

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors.





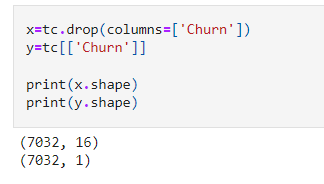
**\*\* Phone service and senior citizen has some outliers so we will drop these two columns**



**\*\* No outliers are present now**

We will create variables for the input variables and target variable

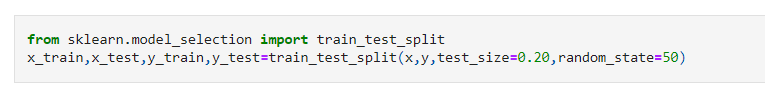
‘x’ Is for input variables so we’ll drop the column ‘Churn’ for ‘x’ & ‘y’ variable is for our target variable which is churn, so we’ll only include ‘churn’ in ‘y’.



‘x’ has 7032 rows and 16 columns while ‘y’ also has 7032 rows but only 1 column.

Now we will train our data so it can learn the independent variables and able to predict pattern of the output variable.

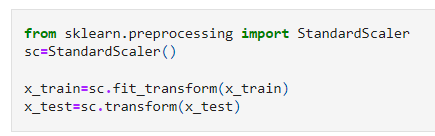
Importing train\_test\_split from sklearn.model\_selection library. Giving the variable ‘x’ and ‘y’ to the train\_test\_split. We’re going to set test size 20 % and training 80%.



**STANDARD SCALER**

Python sklearn library offers us with StandardScaler() function to standardize the data values into a standard format.

Importing StandardScaler() from sklearn.preprocessing and creating a variable to store the StandardScaler(), then scaling the x\_train and x\_test



**MACHINE LEARNING MODELS**

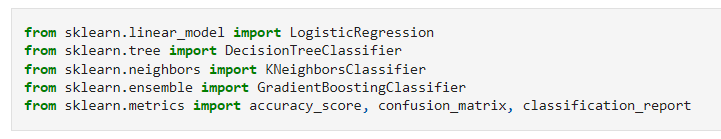
A machine learning model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

Importing different algorithms from different libraries, these are : -

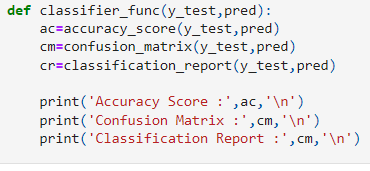
* LogisticRegression from sklearn.linear\_model library
* DecisionTreeClassifier from sklearn.tree library
* KNeighborsClassifier from sklearn.neighbors library
* GradientBoostingClassifier from sklearn.ensemble library

Importing metrics from library sklearn.metrics, these are : -

* accuracy\_score
* confusion\_matrix
* classification\_report



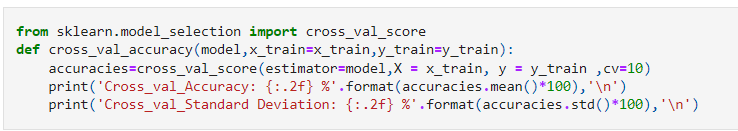
1. Creating function named classifier\_func for checking accuracy score, confusion matrix and classification report of prediction models
2. Creating separate variables for all the metrices, ‘ac’ for accuracy score, ‘cm’ **for** confusion\_matrix, ‘cr’ for classification\_report.
3. Also printing the scores of each metrices.



**CROSS VALIDATION SCORE**

Cross-validation starts by shuffling the data (to prevent any unintentional ordering errors) and splitting it into k folds. Then k models are fit on k−1k of the data (called the training split) and evaluated on 1k of the data (called the test split). The results from each evaluation are averaged together for a final score, then the final model is fit on the entire dataset for operationalization.

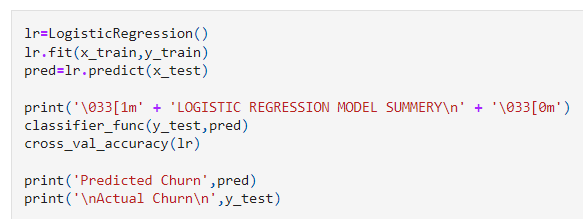
* We’ll make a function named cross\_val\_accuracy, we’ll give model as estimator, x\_train as X, y\_train as y and cv(iterator) 10.
* Will also print Cross val accuracy and standard deviation

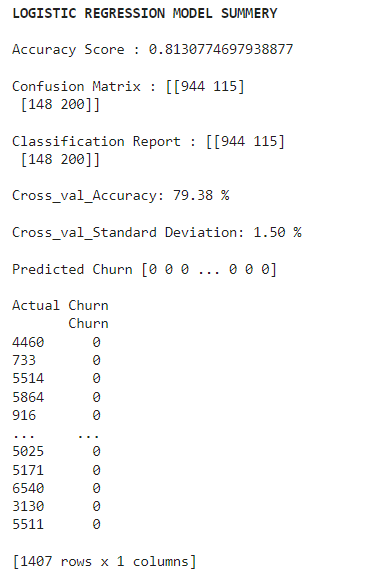


**LOGISTIC REGRESSION**

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

1. **Creating Logistic Regression Model, it'll fit the Logistic Regression model on x\_train and y\_train and predict the x\_test and store it in pred. We will print predicted Churn and actual Churn and outputs of the above functions**
2. Creating variable ‘lr’ for LinearRegression() and fitting x\_train and y\_train. Creating ‘pred’ for predicting x\_test by lr.predict() method.
3. Printing Predicted Churn and Actual Churn and above created classifier\_func and cross\_val\_accuracy.



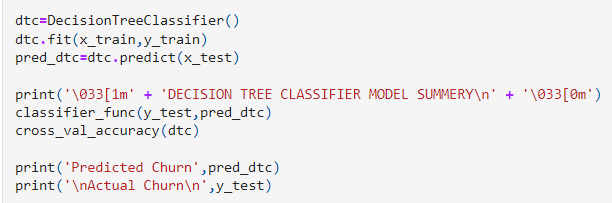


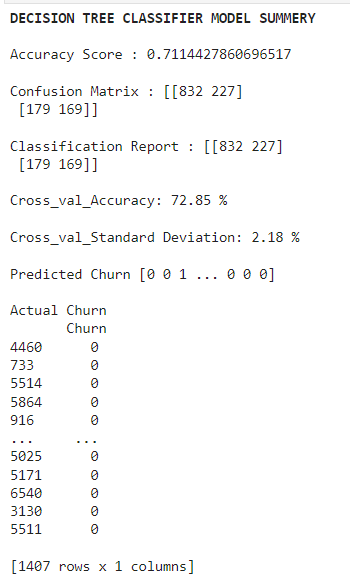
* Logistic Regression model is working very good as it has provided 81.3% accuracy score
* Confusion Matrix : - True Positive – 944, False Positive – 115, False Negative – 148, True Negative – 200.
* Classification Report : - True Positive – 944, False Positive – 115, False Negative – 148, True Negative – 200.
* Cross validation is also very good which is 79.38%, it is showing that we are going good and our model is working fine. But we will still make other machine learning models because who knows they can even perform better than logistic regression model.
* There is very less standard error of 1.5% so this is not a big deal.

**DECISION TREE CLASSIFIER**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems and we’ll also use it as a classifier here. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

1. **Creating Decision Tree Classifier Model, it'll fit the Decision Tree Classifier model on x\_train and y\_train and predict the x\_test and store it in pred\_dtc. We will print predicted Churn and actual Churn and outputs of the above functions**
2. Creating variable ‘dtc’ for DecisionTreeClassifier() and fitting x\_train and y\_train. Creating ‘pred\_dtc’ for predicting x\_test by dtc.predict() method.



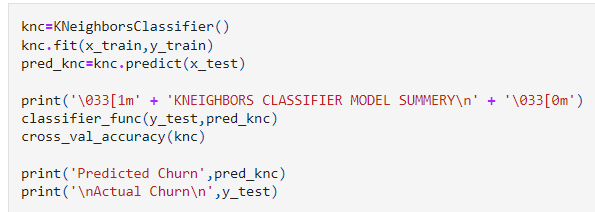


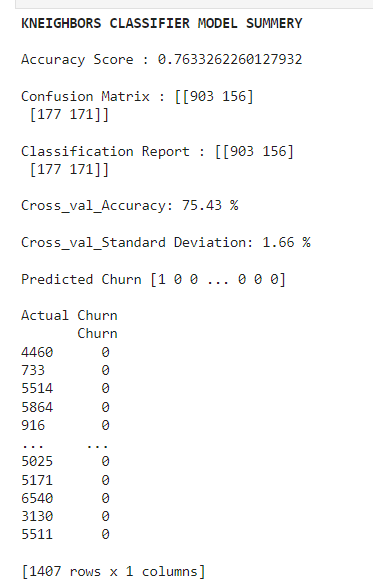
* Decision tree classifier model has provided the accuracy score of 71%, it is also working fine but not as good as Logistic Regression model.
* Confusion Matrix : - True Positive – 832, False Positive – 227, False Negative – 179, True Negative – 169.
* Classification Report : - True Positive – 832, False Positive – 227, False Negative – 179, True Negative – 169.
* There are less true positive and negative metrices.
* Cross validation is also fine which is 72.85 %, it is highly matching with the accuracy score of the model which means this model is going good but we will not prefer this model as our final model because it is not as good as above model.
* There is very less standard error of 2.18% but still we will check some other algorithms also.

**KNEIGHBORS CLASSIFIER**

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

1. **Creating KNeighbors Classifier Model, it'll fit the KNeighbors Classifier model on x\_train and y\_train and predict the x\_test and store it in pred\_knc. We will print predicted Churn and actual Churn and outputs of the above functions**
2. Creating variable ‘knc’ for KNeighborsClassifier() and fitting x\_train and y\_train. Creating ‘pred\_knc’ for predicting x\_test by knc.predict() method.



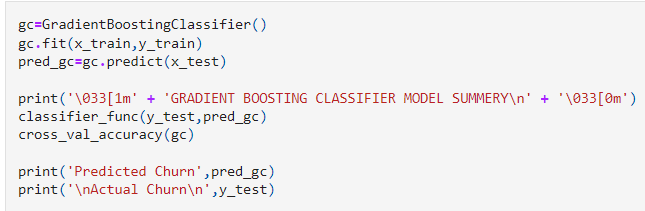


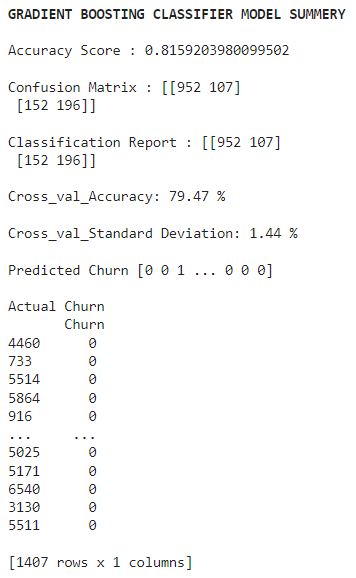
* **KNeighbors** classifier model has provided the accuracy score of 76%, which is better than decision tree classifier model.
* Confusion Matrix : - True Positive – 903, False Positive – 156, False Negative – 177, True Negative – 171.
* Classification Report : - True Positive – 903, False Positive – 156, False Negative – 177, True Negative – 171.
* The true positive values are just little lower than from Logistic regression model. Till now no other model has beaten it.
* Cross validation is also fine which is 75.43 %,. Till now no other model has beaten Logistic Regression.
* There is very less standard error of 1.66%.
* We’ll just check one more algorithm, which is : -

**GRADIENT BOOSTING CLASSIFIER**

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

1. **Creating Gradient Boosting Classifier Model, it'll fit the Gradient Boosting Classifier model on x\_train and y\_train and predict the x\_test and store it in pred\_gc. We will print predicted Churn and actual Churn and outputs of the above functions**
2. Creating variable ‘gc’ for GradientBoostingClassifier() and fitting x\_train and y\_train. Creating ‘pred\_gc’ for predicting x\_test by gc.predict() method.

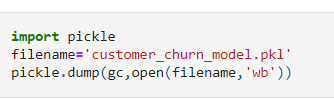




* **GradientBoostingClassifier** model has provided the accuracy score of 81.5%, which is slightly better than Logistic reggression model.
* Confusion Matrix : - True Positive – 952, False Positive – 107, False Negative – 152, True Negative – 196.
* Classification Report : - True Positive – 952, False Positive – 107, False Negative – 152, True Negative – 196.
* The true positive values are just little higher than from Logistic regression model but True negative is lower.
* Cross validation is 79.47% which is best among all the models.
* This model also has the lowest standard error of 1.44%.

**MODEL SAVING**

1. Now we have created some very good machine learning models with different algorithms to predict the churn and we have not got any bad model, all the models were working fine but we have to choose the best performing model from all.
2. **GradientBoostingClassifier is our best performing model, it is predicting churn almost 81.5% accurately. Now we will save this model for future use.**
3. Impoting pickle module, Creating variable ‘filename’ and storing the model with .pkl in it and dumping the model by using pickle so the model can be saved.



**CONCLUSION**

It has become known that predicting churn is one of the most important sources of income to telecom companies. Hence, our research aimed to build a machine learning model that predicts the churn of customer in telecom company. These prediction models need to achieve high accuracy scores and low standard errors. To test and train the model, the data is divided into 80% for training and 20% for testing. We have scaled the x\_train and x\_test to standardize the data values into standard format. We choose to perform cross validation with 10 k-folds. Four algorithms were chosen because of their diversity and applicability in this type of prediction. These algorithms are Linear Regression, Decision Tree, KNeighbors Classifier, Gradient Boosting Classifier. Gradient Boosting Classifier model achieved the best results in all measurements, The accuracy score was 81.5%, Confusion matrix and Classification report of 952 true positive values, standard error of 1.44%. The Logistic regression model comes in second place and KNeighbors Classifier and Decision Tree Classifier came third and fourth regarding accuracy scores. We have saved the Gradient Boosting Classifier for future use when it will be applied in the new set of data as the model will need training each period of time to perform more accurately for bigger datasets.